

โจทย์ปัญหา

โจทย์ 1 (CountPoints) เขียนโปรแกรมเพื่อนับจำนวนจุดบนภาพที่มีค่าเป็น 1 เมื่อ

ข้อมูลนำเข้า บรรทัดแรก คือ ขนาดภาพ กว้างxสูง ($m \times n$) บรรทัดถัดไป m บรรทัด คือ ค่าข้อมูลในแต่ละแถว ซึ่งมีค่าเป็น 0 หรือ 1

ข้อมูลนำออก คือ จำนวนตำแหน่งที่มีค่า 1

```
/**
Example: Counting points
Input : 3 3          line 1 size row x column
        1 0 1        data of row 1
        0 1 0        data of row 2
        0 0 0        data of row 3
Output : 3
*/
```

โจทย์ 2 (GetPoints2Arr) เขียนโปรแกรมเพื่อแสดงตำแหน่งพิกัดบนภาพที่มีค่าเป็น 1 เมื่อ

ข้อมูลนำเข้า บรรทัดแรก คือ ขนาดภาพ กว้างxสูง ($m \times n$) บรรทัดถัดไป m บรรทัด คือ ค่าข้อมูลในแต่ละแถว ซึ่งมีค่าเป็น 0 หรือ 1 เว้นระยะระหว่างค่า 1 space

ข้อมูลนำออก คือ ลำดับตำแหน่งของภาพ (x,y) ในระบบพิกัดภาพบนหน้าจอ (ไล่ลำดับตามแถวและคอลัมน์ของข้อมูลนำเข้า) เว้นระยะระหว่างจุด 1 space

Note: ระบบพิกัดภาพของหน้าจอ มุมบนซ้าย คือ จุด 0,0 แนวนอน คือ ความกว้างหรือแกน x และแนวตั้ง คือ ความสูงหรือแกน y

```
/**
Problem: show all points
Input : 3 3          line 1 size of input row x column
        1 0 1        data of row 1
        0 1 0        data of row 2
        0 1 0        data of row 3
Output : (0,0) (2,0) (1,1) (1,2)   ตำแหน่ง x,y ของจุด
*/
```

row 1

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int m, n;
6     cin >> m >> n;
7     int arr[m][n];
8     int a = 0;
9     for(int i = 0; i < m; i++){
10         for(int j = 0; j < n; j++){
11             cin >> arr[i][j];
12         }
13     }
14
15     for(int i = 0; i < m; i++){
16         for(int j = 0; j < n; j++){
17             if(arr[i][j] == 1){
18                 a++;
19             }
20         }
21     }
22
23     cout << a;
24     return 0;
25 }
```

row 2

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int m, n;
6     cin >> m >> n;
7     int arr[m][n];
8
9     for(int i = 0; i < m; i++){
10         for(int j = 0; j < n; j++){
11             cin >> arr[i][j];
12         }
13     }
14
15     for(int i = 0; i < m; i++){
16         for(int j = 0; j < n; j++){
17             if(arr[i][j] == 1){
18                 cout << "(" << j << "," << i << ")" << ' ';
19             }
20         }
21     }
22     return 0;
23 }
```

โจทย์ 3 (CheckTri) เขียนโปรแกรมเพื่อตรวจสอบตำแหน่งจุดสามจุดบนภาพที่มีค่าเป็น 1 ว่าเป็นจุดที่สามารถประกอบกันเป็นสามเหลี่ยมมุมฉากหรือไม่ เมื่อ

ข้อมูลนำเข้า บรรทัดแรก คือ ขนาดภาพ กว้างxสูง ($m \times n$) บรรทัดถัดไป m บรรทัด คือ ค่าข้อมูลในแต่ละแถว ซึ่งมีค่าเป็น 0 หรือ 1

ข้อมูลนำออก คือ รายการจุดทั้งสาม และผลการตรวจสอบ (YES/NO) มี space ระหว่างค่า 1 space

```
/**
Example1:
Input: 3 3          line 1: size row x column
      1 0 1          data of row 1
      0 1 0          data of row 2
      0 0 0          data of row 3
Output: (0,0) (2,0) (1,1) NO

Example1:
Input: 3 4          line 1: size row x column
      1 0 1          data of row 1
      0 0 0          data of row 2
      1 0 0          data of row 3
      0 0 0          data of row 4
Output: (0,0) (2,0) (0,2) YES
*/
```

โจทย์ 4 (RectInside) ให้จุดมุมของสี่เหลี่ยมมุมฉากมา 2 รูป จงเขียนโปรแกรมเพื่อตรวจสอบว่าสี่เหลี่ยมสองรูปนั้นซ้อนทับกันแบบสมบูรณ์หรือไม่ เมื่อ

ข้อมูลนำเข้า บรรทัดแรก คือ คู่อันดับของจุด 4 จุดของสี่เหลี่ยมรูปแรก บรรทัดที่สอง คือ คู่อันดับของจุด 4 จุดของสี่เหลี่ยมรูปที่สอง โดยที่ลำดับจุดการนำเข้าเป็นแบบวนขวา

ข้อมูลนำออก คือ ผลการตรวจสอบ (YES/NO)

```
/**
Example 1:
Input:
      0 0 0 4 4 4 4 0
      1 1 1 3 3 3 3 1
Output: YES

Example 2:
Input:
      0 0 0 4 4 4 4 0
      1 1 1 6 3 6 3 1
Output: NO
*/
```

Ques 9

```
1 #include <stdio.h>
2
3 int main(){
4     int a[8],b[8],i;
5     for(i=0;i<8;i++){
6         scanf("%d",&a[i]);
7     }
8     for(i=0;i<8;i++){
9         scanf("%d",&b[i]);
10    }
11    if((a[4]-a[2])/(a[5]-a[7])==(b[4]-b[2])/(b[5]-b[7])) printf("YES");
12    else printf("NO");
13 }
```

โจทย์ 5 (InsidePartInde) ให้จุดมุมของสี่เหลี่ยมมุมฉากมา 2 รูป จงเขียนโปรแกรมเพื่อตรวจสอบว่าสี่เหลี่ยมสองรูปนั้นซ้อนทับกันบางส่วน (Partial) หรืออิสระกัน (Independent) หรืออยู่ภายในอีกรูปหนึ่ง (Inside) เมื่อ

ข้อมูลนำเข้า บรรทัดแรก คือ คู่อันดับของจุด 4 จุดของสี่เหลี่ยมรูปแรก บรรทัดที่สอง คือ คู่อันดับของจุด 4 จุดของสี่เหลี่ยมรูปที่สอง โดยที่ลำดับจุดการนำเข้าเป็นแบบวนขวา

ข้อมูลนำออก คือ ผลการตรวจสอบ (Inside/Partial/Independent)

/**

Example 1:

Input:

0 0 0 4 4 4 4 0

พิกัด x,y ของจุดทั้งสี่ของรูปแรก

1 1 1 3 3 3 3 1

พิกัด x,y ของจุดทั้งสี่ของรูปที่สอง

Output: Inside

Example 2:

Input:

0 0 0 4 4 4 4 0

1 1 1 6 3 6 3 1

Output: Partial

Example 3:

Input:

0 0 0 4 4 4 4 0

7 1 7 3 10 3 10 1

Output: Independent

*/

โจทย์ 6 (ImageRotate) จงเขียนโปรแกรมเพื่อหมุนภาพต้นฉบับขนาด $n \times n$ pixels ทำมุม 90 องศา ไปทางขวา (R) หรือซ้าย (L) ตามจำนวนครั้ง (steps) ที่กำหนด โดยแสดงภาพผลลัพธ์ที่ได้หลังจากหมุน ดังตัวอย่าง

/**

Example 1:

Input: 5 line 1 size $n \times n$ of image
 0 0 0 0 0 data of row 1
 0 1 0 0 0 data of row 2
 0 1 1 0 0 data of row 3
 0 1 1 1 0 data of row 4
 0 0 0 0 0 data of row 5
 R 1 rotate 90 degree right 1 step

Output:

 0 0 0 0 0 data of row 1
 0 1 1 1 0 data of row 2
 0 1 1 0 0 data of row 3
 0 1 0 0 0 data of row 4
 0 0 0 0 0 data of row 5

Example 2:

Input: 5 line 1 size $n \times n$ of image
 0 0 0 0 0 data of row 1
 0 1 0 0 0 data of row 2
 0 1 1 0 0 data of row 3
 0 1 1 1 0 data of row 4
 0 0 0 0 0 data of row 5
 L 1 rotate 90 degree left 1 step

Output:

 0 0 0 0 0 data of row 1
 0 0 0 1 0 data of row 2
 0 0 1 1 0 data of row 3
 0 1 1 1 0 data of row 4
 0 0 0 0 0 data of row 5

mod

*/

โจทย์ 7 (MoveImg) จงเขียนโปรแกรมเพื่อย้าย (move) ภาพต้นฉบับไปทางซ้าย-ขวา (LR) หรือ บน-ล่าง (UD) ของเฟรมแสดงผลตามที่กำหนด

หากค่า LR เป็น + หมายถึงย้ายไปทางขวาตามจำนวนหน่วยที่ระบุ และหากเป็น - หมายถึง ย้ายไปทางซ้ายตามจำนวนหน่วยที่ระบุ

หากค่า UD เป็น + หมายถึงย้ายลงด้านล่างตามจำนวนหน่วยที่ระบุ และหากเป็น - หมายถึง ย้ายขึ้นด้านบนตามจำนวนหน่วยที่ระบุ

การย้ายตำแหน่งภาพไปทางซ้าย-ขวา/บน-ล่าง นั้น ภาพจะต้องอยู่ภายในขอบเฟรม หมายความว่าถ้าชนเฟรมแล้วให้หยุดแค่นั้น ดังตัวอย่าง

/**

Example 1:

Input: 5 7	line 1 size of data
0 0 0 0 0 0 0	data of row 1
0 1 0 0 0 0 0	data of row 2
0 1 1 0 0 0 0	data of row 3
0 1 1 1 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5
-2 1	move LR:-2 UD:1

Output:

0 0 0 0 0 0 0	data of row 1
0 0 0 0 0 0 0	data of row 2
1 0 0 0 0 0 0	data of row 3
1 1 0 0 0 0 0	data of row 4
1 1 1 0 0 0 0	data of row 5

Example 2:

Input: 5 7	line 1 size of data
0 0 0 0 0 0 0	data of row 1
0 1 0 0 0 0 0	data of row 2
0 1 1 0 0 0 0	data of row 3
0 1 1 1 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5
2 -1	move LR:2 UD:-1

Output:

0 0 0 1 0 0 0	data of row 1
0 0 0 1 1 0 0	data of row 2
0 0 0 1 1 1 0	data of row 3
0 0 0 0 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5

*/

โจทย์ 8 จงเขียนโปรแกรมเพื่อขยับ (shift) ภาพต้นฉบับไปทางซ้ายหรือขวาของเฟรมแสดงผลตามที่กำหนด การขยับซ้ายหรือขวานั้น หากภาพสุดขอบเฟรมให้วนกลับมาแสดงต่อยังขอบอีกด้าน ดังตัวอย่าง

```
/**
Example 1:
Input: 5 7                                line 1 size of data
      0 0 0 0 0 0 0                        data of row 1
      0 1 0 0 0 0 0                        data of row 2
      0 1 1 0 0 0 0                        data of row 3
      0 1 1 1 0 0 0                        data of row 4
      0 0 0 0 0 0 0                        data of row 5
      L 2                                shift left 2 steps

Output:
      0 0 0 0 0 0 0                        data of row 1
      0 0 0 0 0 0 1                        data of row 2
      1 0 0 0 0 0 1                        data of row 3
      1 1 0 0 0 0 1                        data of row 4
      0 0 0 0 0 0 0                        data of row 5

Example 2:
Input: 5 7                                line 1 size of data
      0 0 0 0 0 0 0                        data of row 1
      0 1 0 0 0 0 0                        data of row 2
      0 1 1 0 0 0 0                        data of row 3
      0 1 1 1 0 0 0                        data of row 4
      0 0 0 0 0 0 0                        data of row 5
      R 1                                shift right 1 step

Output:
      0 0 0 0 0 0 0                        data of row 1
      0 0 1 0 0 0 0                        data of row 2
      0 0 1 1 0 0 0                        data of row 3
      0 0 1 1 1 0 0                        data of row 4
      0 0 0 0 0 0 0                        data of row 5
*/
```

ข้อมูลนำเข้า บรรทัดที่ 1 ขนาดความสูงและความกว้างของหน้าจอแสดงผล (screen) แถว (m) คูณ คอลัมน์ (n)

m แถวถัดมา คือ ค่าข้อมูลในแต่ละคอลัมน์ ซึ่งประกอบด้วย 0 หรือ 1 ซึ่งคั่นด้วยที่ว่าง 1 ตำแหน่ง

บรรทัดสุดท้าย คือ รูปแบบการขยับภาพ (L/R) และจำนวนครั้งของการขยับ (เป็นจำนวนเต็มบวก ≤ 1000) เช่น L 2 คือ ขยับไปทางซ้าย 2 ตำแหน่ง, R 1 คือ ขยับไปทางขวา 1 ตำแหน่ง

ข้อมูลนำออก คือ ผลลัพธ์หน้าจอแสดงผลหลังดำเนินการขยับภาพตามที่กำหนด ซึ่งมีขนาดเท่ากับ m คูณ n เช่นเดิม

หมายเหตุ การใช้เศษจากการหาร (modulo) จะช่วยให้เขียนโปรแกรมง่ายขึ้น

โจทย์ 9 (RectOrTri) จงเขียนโปรแกรมเพื่อตรวจสอบภาพว่าเป็นรูปสามเหลี่ยม (Triangle) หรือสี่เหลี่ยม (Rectangle) เมื่อกำหนดตำแหน่งจุดของเส้นบนภาพเป็น 1 (สีดำ) ตำแหน่งที่เป็น 0 เป็นพื้นขาว ดังตัวอย่าง

/**

Example 1:

Input: 5 7	line 1 size of data m x n
0 0 0 0 0 0 0	data of row 1
0 1 1 1 1 1 0	data of row 2
0 1 0 0 0 1 0	data of row 3
0 1 1 1 1 1 0	data of row 4
0 0 0 0 0 0 0	data of row 5

Output:

Rectangle

Example 2:

Input: 5 7	line 1 size of data m x n
0 0 0 0 0 0 0	data of row 1
0 1 0 0 0 0 0	data of row 2
0 1 1 0 0 0 0	data of row 3
0 1 1 1 0 0 0	data of row 4
0 0 0 0 0 0 0	data of row 5

Output:

Triangle

Example 3:

Input: 5 7	line 1 size data m x n
0 0 0 0 0 1 0	data of row 1
0 0 0 0 1 1 0	data of row 2
0 0 0 1 0 1 0	data of row 3
0 0 1 0 0 1 0	data of row 4
0 1 1 1 1 1 0	data of row 5

Output:

Triangle

*/